

Exercice « Gestion de collections d'automobiles dans MongoDB» **(06.05.21)**

Pour les besoins de cette exercice, seul le Mongo Shell a été employé pour résoudre aux questions suivantes

Question 1 : Insérer les documents dans une collection Voiture

Dans le cadre de cette question, les documents ont été insérés manuellement via le terminal Mongo Shell.

commandes exécutées :

```
db.voitures.insertOne({marque : 'ferrari', modele : 'f40', denomination : 'testarossa',  
cylindree : '2946', couleur : 'rouge', proprietaire : 'valerio'})
```

```
db.voitures.insertOne({marque : 'lamborghini', modele : 'countach', denomination : 'KITT',  
cylindree : '5167', couleur : 'jaune', proprietaire : 'rafik'})
```

```
db.voitures.insertOne({marque : 'de tomaso', modele : 'f', denomination : '', cylindree : '5800',  
couleur : 'noir', proprietaire : 'rafik'})
```

```
db.voitures.insertOne({marque : 'pontiac', modele : 'firebird', denomination : 'KITT',  
cylindree : '5700', couleur : 'noir', proprietaire : 'valerio'})
```

```
db.voitures.insertOne({marque : 'volkswagen', modele : 'coccinelle',  
denomination : 'choupette', cylindree : '1584', couleur : 'blanc', proprietaire : 'laurent'})
```

```
db.voitures.insertOne({marque : 'ford', modele : 'mustang', denomination : 'eleanor',  
cylindree : '7000', couleur : 'blanc', proprietaire : 'laurent'})
```

```
db.voitures.insertOne({marque : 'chevrolet', modele : 'corvette', denomination : 'lola',  
cylindree : '5400', couleur : 'rouge', proprietaire : 'valerio'})
```

```
db.voitures.insertOne({marque : 'cadillac', modele : 'eldorado', denomination : 'ectomobile-  
1', cylindree : '6400', couleur : 'blanc', proprietaire : 'laurent'})
```

```
db.voitures.insertOne({marque : 'dodge', modele : 'charger', denomination : 'general lee',  
cylindree : '7206', couleur : 'orange', proprietaire : 'rafik'})
```

question 2 : Selectionner tous les documents Selectionner tous les documents avec Valerio ou Rafik en propriétaire

commande exécutée : `db.voitures.find(`
 `{proprietaire: {`
 `$in: ['rafik', 'valerio']`
 `} }`
`)`

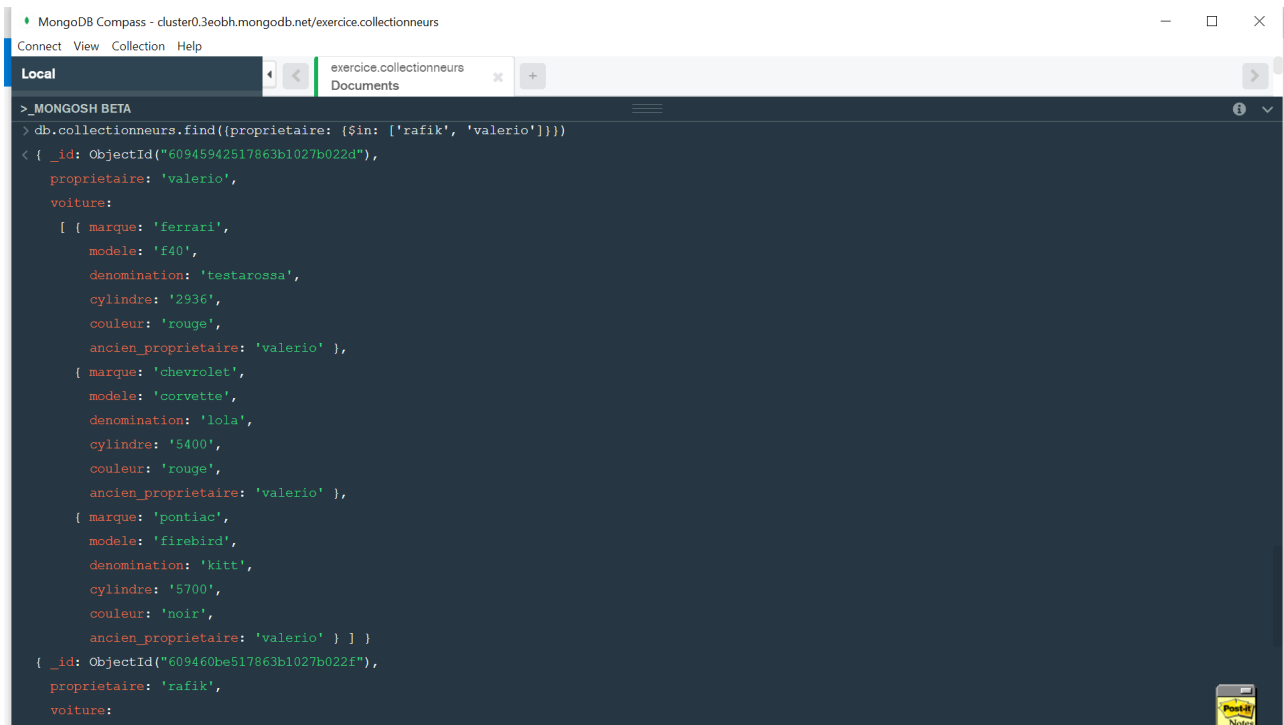


Illustration 1: Sortie des résultats après exécution des instructions

question3 : Mettre à jour tous les documents avec Valerio en propriétaire :
Ajouter un champ PropriétairePrecedent auquel vous donnerez la valeur
Valerio Mettre à jour la valeur du champ Propriétaire avec votre prénom

commande exécutée :

```
db.voitures.updateMany({proprietaire:'valerio'},
    { $set: {ancien_proprietaire:'valerio',
              proprietaire:'pw'
            }
    }
)
```

MongoDB Compass - cluster0.3eobh.mongodb.net/exercice.voitures

Connect View Collection Help

Local

5 DBS 12 COLLECTIONS

☆ FAVORITE

HOSTS

cluster0-shard-00-01.3eobh.mongodb.net

cluster0-shard-00-00.3eobh.mongodb.net

cluster0-shard-00-02.3eobh.mongodb.net

CLUSTER

Replica Set (atlas-vrhcg-s...

3 Nodes

EDITION

MongoDB 4.4.5 Enterprise

Filter your data

+

exercice.voitures Documents

exercice.voitures

DOCUMENTS 0 TOTAL SIZE 0B AVG. SIZE 0B

INDEXES 1 TOTAL SIZE 4.0KB AVG. SIZE 4.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

ADD DATA VIEW

Displaying documents 1 - 10 of 10

REFRESH

```
{
  "_id": ObjectId("6093ff73517863b1027b0228"),
  "marque": "chevrolet",
  "modele": "corvette",
  "denomination": "lola",
  "cylindree": "5400",
  "couleur": "rouge",
  "proprietaire": "pw",
  "ancien_proprietaire": "valerio"
}
```

> MONGOSH BETA

TypeError: db.voitures.find(...).updateMany is not a function

```
> db.voitures.updateMany({proprietaire:'valerio'}, { $set: {ancien_proprietaire:'valerio', proprietaire:'pw'}})
< { acknowledged: true,
  insertedId: null,
  matchedCount: 3,
  modifiedCount: 3,
  upsertedCount: 0 }
> db.voitures.updateMany({proprietaire:'valerio'}, { $set: {ancien_proprietaire:'valerio', proprietaire:'pw'}})
< { acknowledged: true,
  insertedId: null,
  matchedCount: 1,
```

Illustration 2: Sortie des résultats après exécution des instructions

question 4 : Supprimer tous les documents ayant "Couleur":"Blanc"

commande exécutée : `db.voitures.deleteMany({couleur:'blanc'})`

MongoDB Compass - cluster0.3eobh.mongodb.net/exercice.voitures

Connect View Collection Help

Local

5 DBS 12 COLLECTIONS

☆ FAVORITE

HOSTS

cluster0-shard-00-01.3eobh.mongodb.net

cluster0-shard-00-00.3eobh.mongodb.net

cluster0-shard-00-02.3eobh.mongodb.net

CLUSTER

Replica Set (atlas-vrhcg-shard-0)

3 Nodes

EDITION

MongoDB 4.4.5 Enterprise

Filter your data

admin

config

exercice

voitures

local

+

exercice.voitures Documents

Documents Aggregations Schema Explain Plan Indexes Validation

DOCUMENTS 0 TOTAL SIZE 0B AVG. SIZE 0B INDEXES 1 TOTAL SIZE 4.0KB AVG. SIZE 4.0KB

Documents

FILTER { field: 'value' }

ADD DATA VIEW

Displaying documents 1 - 10 of 10 REFRESH

ancien_proprietaire: "valerio"

_id: ObjectId("6093ff69517863b1027b0226")
marque: "volkswagen"
modele: "coccinelle"
denomination: "chouquette"
cylindree: "1584"
couleur: "blanc"
proprietaire: "laurent"

_id: ObjectId("6093ff6e517863b1027b0227")
marque: "ford"
modele: "mustang"
denomination: "eleanor"
cylindree: "7000"
couleur: "blanc"
proprietaire: "laurent"

> _MONGOSH BETA

```
insertOne({
  marque: 'volkswagen',
  modele: 'coccinelle',
  denomination: 'chouquette',
  cylindre: '1584',
  couleur: 'blanc'
})
{
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Enterprise atlas-vrhcg-shard-0 [primary] > db.voitures.deleteMany({couleur: 'blanc'})

Illustration 3: Sortie des résultats après exécution des instructions

question 5: Insérer les documents dans une collection "Collectionneur"

commande exécutée :

```
db.collectionneurs.insertOne({proprietaire : 'laurent', voiture:[ { marque : 'volkswagen',
modele : 'coccinelle', denomination : 'chouquette', cylindre : '1584', couleur : 'blanc'},
{marque : 'cadillac', modele : 'eldorado', denomination : 'ectomobile-1', cylindre : '6400',
couleur : 'blanc'} ] })
```

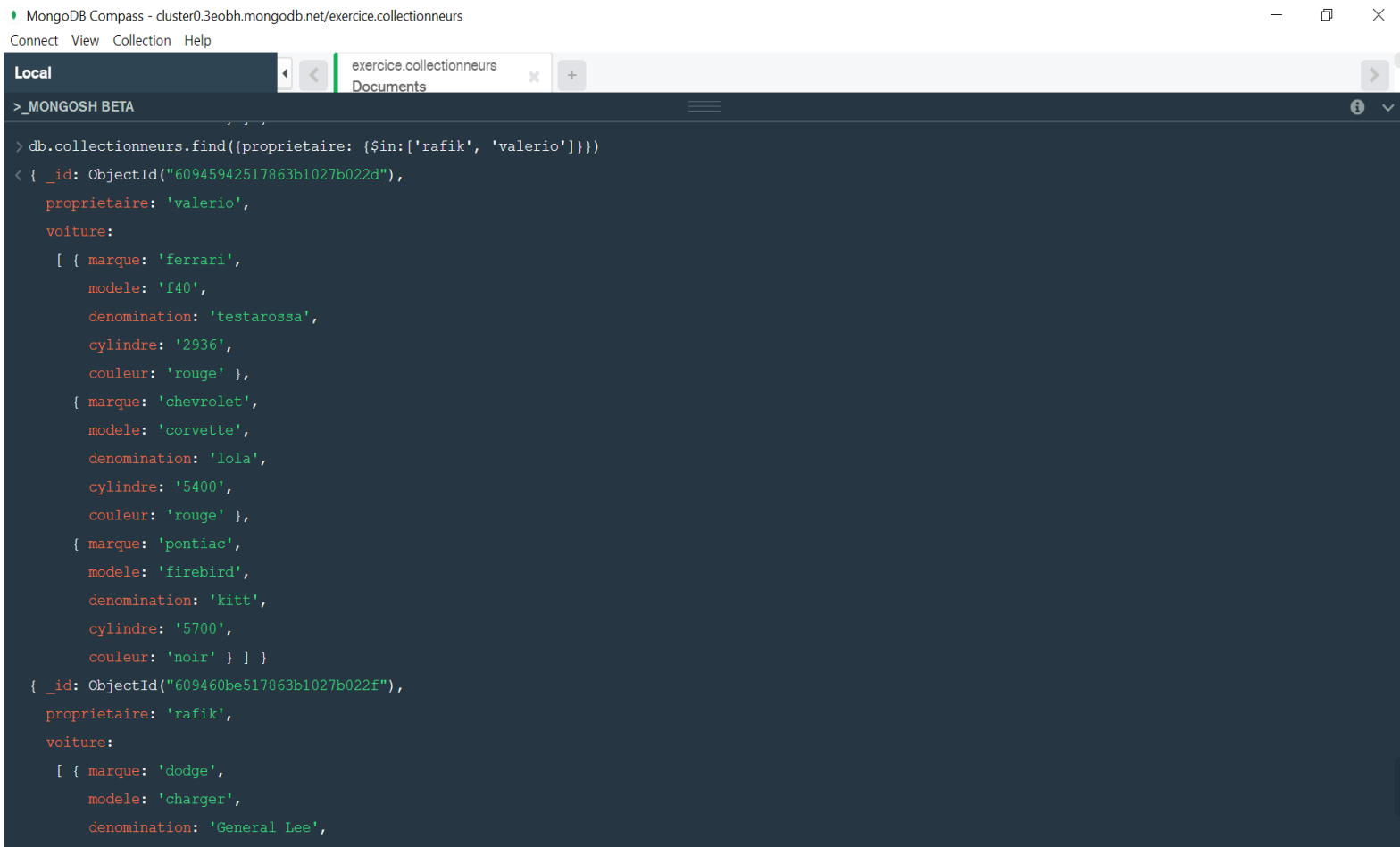
```
db.collectionneurs.insertOne( { proprietaire : 'valerio', 'voiture' : [{ marque : 'ferrari',
modele : 'f40', denomination : 'testarossa', cylindre : '2936', couleur : 'rouge'}
, { marque : 'chevrolet', modele : 'corvette', denomination : 'lola', cylindre : '5400',
couleur : 'rouge'}, { marque : 'pontiac', modele : 'firebird', denomination : 'kitt', cylindre :
'5700', couleur : 'noir'} ]} )
```

```
db.collectionneurs.insertOne({proprietaire : 'rafik', voiture : [ {marque : 'dodge', modele :
'charger', denomination : 'General Lee', cylindre : '7206', couleur : 'orange'}, {marque :
'lamborghini', modele : 'countach', denomination : '', cylindre : '5167', couleur : 'Jaune'},
```

```
{marque : 'de tomaso', modele : 'countach', denomination : '', cylindre : '5167',  
couleur : 'jaune'}, {marque : 'ford', modele : 'mustang', denomination : 'eleanor', cylindre :  
'7000', couleur : 'gris'} ] } )
```

6- Selectionner tous les documents avec Valerio ou Rafik en propriétaire

commande entrée : `db.collectionneurs.find({proprietaire:
{$in:['rafik', 'valerio']}})`



```
> db.collectionneurs.find({proprietaire: {$in:['rafik', 'valerio']}})  
< { _id: ObjectId("60945942517863b1027b022d"),  
  proprietaire: 'valerio',  
  voiture: [ { marque: 'ferrari',  
    modele: 'f40',  
    denomination: 'testarossa',  
    cylindre: '2936',  
    couleur: 'rouge' },  
    { marque: 'chevrolet',  
      modele: 'corvette',  
      denomination: 'lola',  
      cylindre: '5400',  
      couleur: 'rouge' },  
    { marque: 'pontiac',  
      modele: 'firebird',  
      denomination: 'kitt',  
      cylindre: '5700',  
      couleur: 'noir' } ] }  
{ _id: ObjectId("609460be517863b1027b022f"),  
  proprietaire: 'rafik',  
  voiture: [ { marque: 'dodge',  
    modele: 'charger',  
    denomination: 'General Lee',
```

Illustration 4: Sortie des résultats après exécution des instructions

7- Mettre à jour tous les documents avec Valerio en propriétaire : Ajouter un champ PropriétairePrécédent auquel vous donnerez la valeur Valerio Mettre à jour la valeur du champ Propriétaire avec votre prénom

commande entrée : `db.collectionneurs.updateMany({'proprietaire':'valerio'}, {$set:
{'proprietaire': 'john', 'voiture.$[].ancien_proprietaire':'valerio'} })`

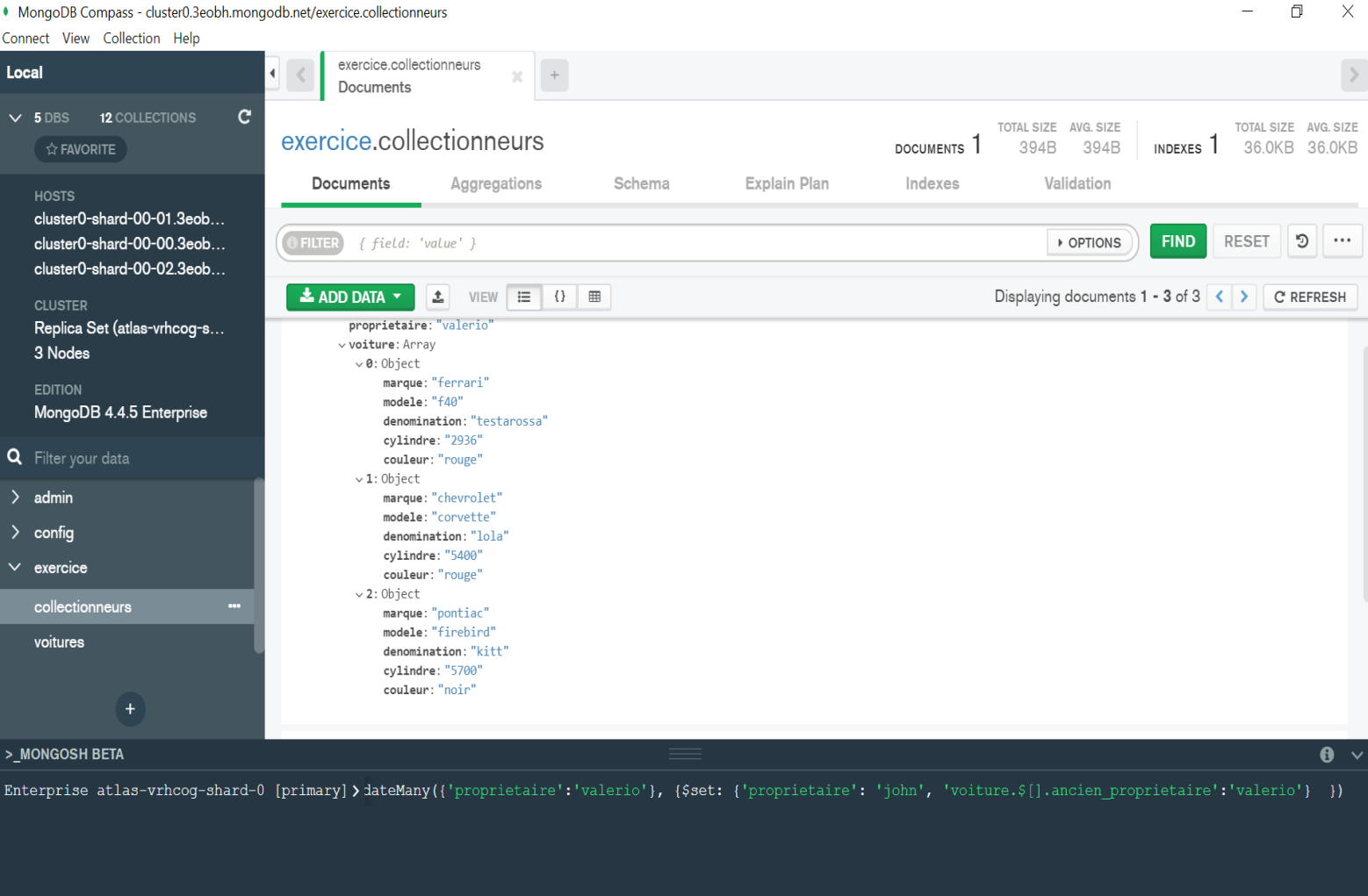


Illustration 5: Dataset avant exécution de la commande

MongoDB Compass - cluster0.3eobh.mongodb.net/exercice.collectionneurs

Connect View Collection Help

Local

5 DBS 12 COLLECTIONS

☆ FAVORITE

HOSTS

- cluster0-shard-00-01.3eob...
- cluster0-shard-00-02.3eob...
- cluster0-shard-00-03.3eob...

CLUSTER

Replica Set (atlas-vrhcg-s...

3 Nodes

EDITION

MongoDB 4.4.5 Enterprise

Filter your data

- admin
- config
- exercice
- collectionneurs
- voitures

> _MONGOSH BETA

```
> db.collectionneurs.updateMany({'proprietaire':'valerio'}, {$set: {'proprietaire': 'john', 'voiture.$[].ancien_proprietaire':'valerio'}})
< { acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
}
```

exercice.collectionneurs Documents

DOCUMENTS 1 TOTAL SIZE 394B AVG. SIZE 394B INDEXES 1 TOTAL SIZE 36.0KB AVG. SIZE 36.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

ADD DATA VIEW

Displaying documents 1 - 3 of 3

```
{
  "proprietaire": "john",
  "voiture": [
    {
      "marque": "ferrari",
      "modele": "f40",
      "denomination": "testarossa",
      "cylindre": "2936",
      "couleur": "rouge",
      "ancien_proprietaire": "valerio"
    },
    {
      "marque": "chevrolet",
      "modele": "corvette",
      "denomination": "lola",
      "cylindre": "5400",
      "couleur": "rouge",
      "ancien_proprietaire": "valerio"
    },
    {
      "marque": "pontiac",
      "modele": "firebird",
      "denomination": "kitt",
      "cylindre": "5700",
      "couleur": "blanc",
      "ancien_proprietaire": "valerio"
    }
  ]
}
```

Illustration 6: Dataset après exécution de la commande

8- Supprimer tous les documents ayant "Couleur":"Blanc"

```
db.collectionneurs.updateMany({}, {$pull: {'voiture':{'couleur':'blanc'}}}, {multi:true})
```

MongoDB Compass - cluster0.3eobh.mongodb.net

Connect View Collection Help

Local

5 DBS 12 COLLECTIONS

☆ FAVORITE

HOSTS

- cluster0-shard-00-01.3eob...
- cluster0-shard-00-02.3eob...
- cluster0-shard-00-03.3eob...

CLUSTER

Replica Set (atlas-vrhcg-s...

3 Nodes

EDITION

MongoDB 4.4.5 Enterprise

Filter your data

- admin
- config
- exercise
 - collectionneurs

+

> _MONGOSH BETA

exercise.collectionneurs Documents

DOCUMENTS 3 TOTAL SIZE 1.3KB AVG. SIZE 431B INDEXES 1 TOTAL SIZE 36.0KB AVG. SIZE 36.0KB

exercise.collectionneurs

Documents Aggregations Schema Explain Plan Indexes

FILTER { field: 'value' } OPTIONS FIND RESET ↺ ...

ADD DATA VIEW { } Grid

Displaying documents 1 - 3 of 3 REFRESH

```
{
  "_id": ObjectId("609459f2517863b1027b022e"),
  "proprietaire": "laurent",
  "voiture": Array
}
```

```
{
  "_id": ObjectId("609460be517863b1027b022f"),
  "proprietaire": "rafik",
  "voiture": Array
    0: {
      "marque": "dodge",
      "modele": "charger",
      "denomination": "General Lee",
      "cylindre": "7206",
      "couleur": "orange"
    }
  }
}
```

Past Queries RECENT FAVORITES

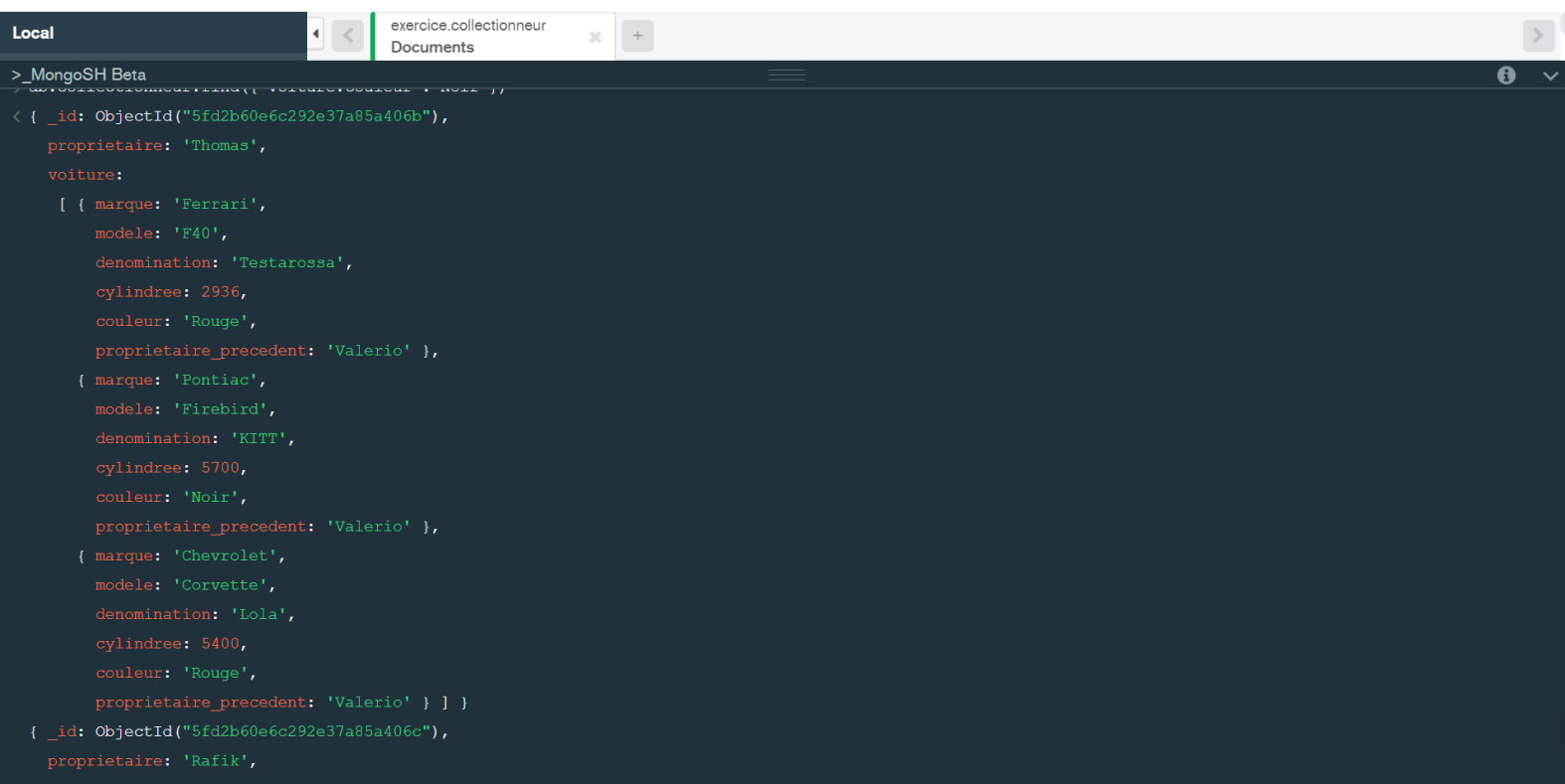
Run a query to see it saved here!

```
> db.collectionneurs.updateMany({}, {$pull: {'voiture':{'couleur':'blanc'}}}, {multi:true})
< { acknowledged: true,
  insertedId: null,
  matchedCount: 3,
  modifiedCount: 1,
  upsertedCount: 0 }
```

Illustration 7: Dataset après exécution de la commande

9- Selectionner tous les documents ayant "Couleur":"Noir", et remarquez le détail du résultat. En fonction de la documentation MongoDB, expliquez le résultat et proposez un design différent pour la collection ou un filtre différent.

première entrée des résultats : `db.collectionneur.find({'voiture.couleur':'Noir'})`



```
>_MongoSH Beta
/ exerce.collectionneur Documents
> db.collectionneur.find({'voiture.couleur':'Noir'})
< { _id: ObjectId("5fd2b60e6c292e37a85a406b"),
  proprietaire: 'Thomas',
  voiture:
    [ { marque: 'Ferrari',
      modele: 'F40',
      denomination: 'Testarossa',
      cylindree: 2936,
      couleur: 'Rouge',
      proprietaire_precedent: 'Valerio' },
      { marque: 'Pontiac',
        modele: 'Firebird',
        denomination: 'KITT',
        cylindree: 5700,
        couleur: 'Noir',
        proprietaire_precedent: 'Valerio' },
      { marque: 'Chevrolet',
        modele: 'Corvette',
        denomination: 'Lola',
        cylindree: 5400,
        couleur: 'Rouge',
        proprietaire_precedent: 'Valerio' } ] },
  { _id: ObjectId("5fd2b60e6c292e37a85a406c"),
    proprietaire: 'Rafik',
    voiture:
      [ { marque: 'Ferrari',
        modele: 'F40',
        denomination: 'Testarossa',
        cylindree: 2936,
        couleur: 'Rouge',
        proprietaire_precedent: 'Valerio' },
        { marque: 'Pontiac',
          modele: 'Firebird',
          denomination: 'KITT',
          cylindree: 5700,
          couleur: 'Noir',
          proprietaire_precedent: 'Valerio' },
        { marque: 'Chevrolet',
          modele: 'Corvette',
          denomination: 'Lola',
          cylindree: 5400,
          couleur: 'Rouge',
          proprietaire_precedent: 'Valerio' } ] }
```

Illustration 8: Sortie des résultats après exécution des instructions

Cela s'explique par le format des données. Bien que le passage des données sous un modèle 'dictionnaire : liste de dictionnaires' selon le schéma 'clé_dictionnaire : [{ 'champ1 :valeur1'}, {champ 1 :valeur2}]' semble correcte, la commande tapé ci-dessus renvoie tout document dont un de ses objets contient le mot 'Noir'. Or ces derniers étant organisés en liste, c'est cette dernière qui est renvoyé et non uniquement l'objet concerné. Pour se faire, l'invite de commande suivante peut contourner ce problème, en précisant la nature de ce qui doit être projeté à l'aide du deuxième argument de la commande :

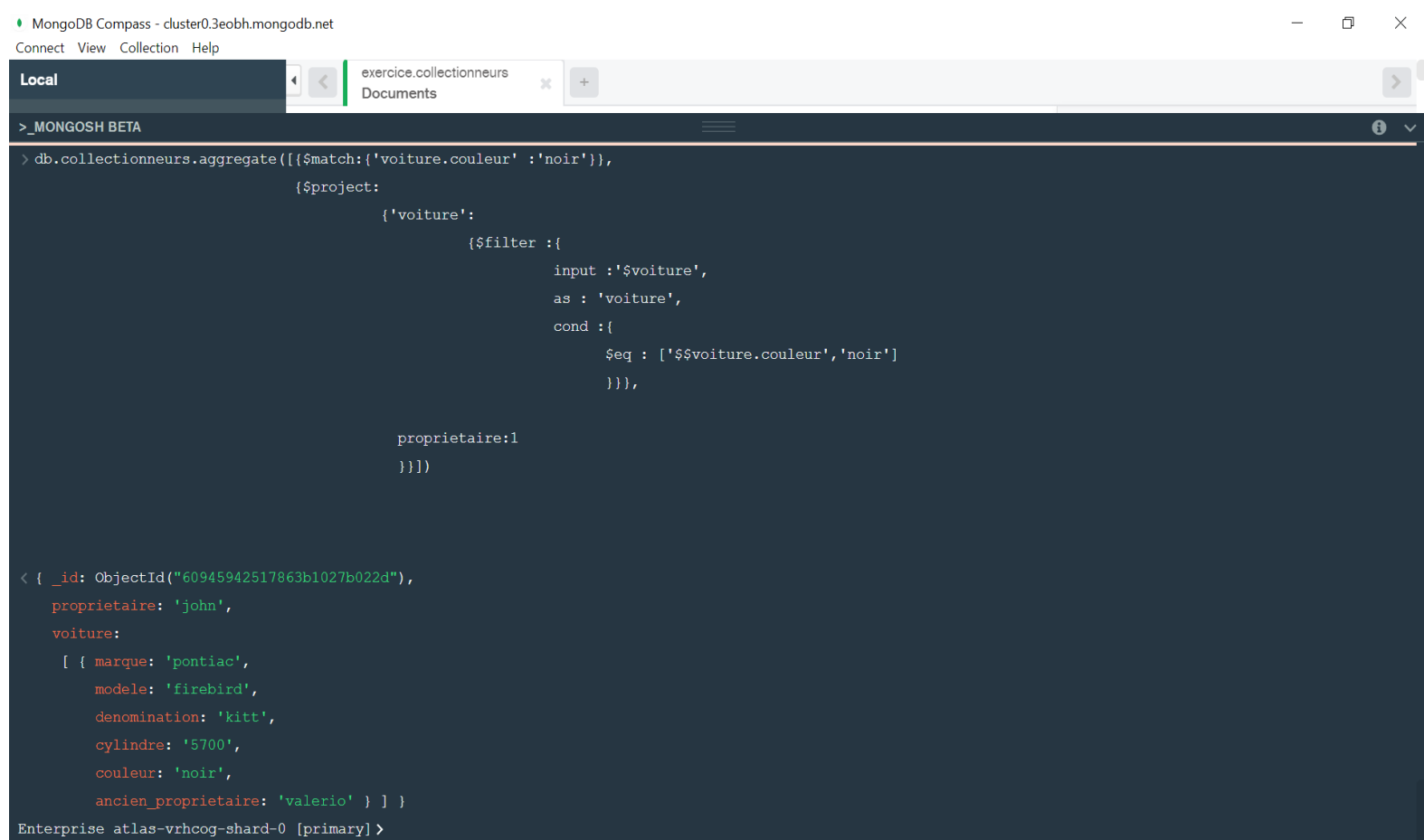
```
db.collectionneurs.aggregate([{$match: {'voiture.couleur' : 'noir'}},
  {$project:
    {'voiture':
      {$filter :{
        input : '$voiture',
        as : 'voiture',
        cond : {
          $eq : ['$voiture.couleur', 'noir']
        }
      }}
    }
  ])
```

```
}}},
```

```
    proprietaire:1  
  }  
}]
```

autre possibilité :

```
db.collectionneurs.find({'voiture.couleur':'orange'}, {'voiture.couleur':1} ).map((doc) =>  
doc.voiture.filter((s) => s.couleur== 'orange'))
```



The screenshot shows the MongoDB Compass web interface. The top bar indicates the connection to 'cluster0.3eobh.mongodb.net'. Below the top bar, there's a tab for 'exercice.collectionneurs' with 'Documents' selected. The main area displays a MongoDB query in the console:

```
> db.collectionneurs.aggregate([{$match: {'voiture.couleur' : 'noir'}},  
    {$project:  
        {'voiture':  
            {$filter : {  
                input : '$voiture',  
                as : 'voiture',  
                cond : {  
                    $eq : [ '$$voiture.couleur', 'noir' ]  
                }  
            }  
        },  
        proprietaire:1  
    }  
])
```

The result of the query is shown below the console:

```
< { _id: ObjectId("60945942517863b1027b022d"),  
  proprietaire: 'john',  
  voiture:  
    [ { marque: 'pontiac',  
      modele: 'firebird',  
      denomination: 'kitt',  
      cylindre: '5700',  
      couleur: 'noir',  
      ancien_proprietaire: 'valerio' } ] }
```

The bottom status bar shows 'Enterprise atlas-vrhcog-shard-0 [primary] >'.

Illustration 9: Sortie des résultats après exécution des instructions